

# パラメータ型オプションを含むコンパイラの最適化オプションセット探索によるプログラムの高速化

Accelerating Program Execution by Exploring Compiler Option Sets Including Parameter Options

後藤潤哉  
Junya Goto

石浦菜岐佐  
Nagisa Ishiura

関西学院大学 理工学部 School of Science and Technology, Kwansei Gakuin University

## 1 はじめに

コンパイラには多くの最適化処理が実装されているが、コンパイラによっては各処理の適用/不適用をオプションで選択できるものがあり、与えられたプログラムに対して、最適なオプションの組み合わせ（オプションセット）を探索する研究が行われている [1]。最適化オプションの中には、最適化処理に関するパラメータを指定するもの（以下「パラメータ型オプション」）も存在する。[2]では、パラメータ型オプションを探索の対象に加えることにより、プログラムのメモリー使用量を削減することを試みているが、効果は得られていない。そこで本研究では、プログラムの実行の高速化を目的として、パラメータ型オプションを含む最適化オプションセットの探索を行うことを提案する。GCC 4.8.1 に対し、5 つのベンチマークで実験を行った結果、本手法により平均約 3.9% の高速化を達成することができた。

## 2 コンパイラの最適化オプション

コンパイラの最適化オプションには、スイッチ型とパラメータ型が存在する。スイッチ型は、最適化処理の適用/不適用を指定するものである。GCC 4.8.1 には 186 個のスイッチ型オプションが存在し、例えば、`-fdse`、`-fno-dse` はそれぞれデッドコード除去の適用/不適用を指定する。これに対し、パラメータ型では、最適化の適用範囲等を数値で指定する。GCC 4.8.1 には 154 個のパラメータ型オプションが存在し、例えば、`--param max-inline-insns-single=100` は、インライン展開の最大命令数を 100 に設定する。

## 3 探索手法

探索アルゴリズムには [2] と同様、シミュレーテッドアニーリングを用いる。スイッチ型とパラメータ型のオプション数をそれぞれ  $n$ 、 $m$  とし、 $i$  番目のパラメータ型オプションの取り得る値の集合を  $D_i$  とする。シミュレーテッドアニーリングの状態は  $(s_1, \dots, s_n, p_1, \dots, p_m)$  (ただし、 $s_i \in \{0,1\}$ 、 $p_i \in D_i$ ) であり、 $s_i$  は  $i$  番目のスイッチ型オプションの指定の有無を、 $p_i$  は  $i$  番目のパラメータ型オプションの値を表す。

状態  $s$  の近傍  $s'$  は、 $s$  の 1 要素の値を変更したものである。 $p_i$  の値をどれだけ変化させるかは温度に依存して変化させる。 $D_i = \{v_1, v_2, \dots, v_k\}$  (ただし、 $v_1 < v_2 < \dots < v_k$ ) であり、 $p_i = v_j$  のとき、 $p_i$  の新しい値は  $v_{j-t}, \dots, v_{j-1}, v_{j+1}, \dots, v_{j+t}$  のいずれかである。ただし、 $t$  は初期温度では  $k/2$  であり、最終温度で 1 となるように、

冷却につれて減少させる。

プログラムの実行時間は UNIX の `time` コマンドを用いて計測し、複数回実行した平均値を用いる。実行回数は初期温度では 1 とし、冷却回数に比例して増加させる。

## 4 実験結果

以上の手法に基づく探索プログラムを Perl 5.14 で実装し、5 つの C プログラムに対して実験を行った。コンパイラは x86 用 GCC 4.8.1 を使用した。全ての最適化オプションから、エラーや警告が発生するものを除き、スイッチ型 164 個、パラメータ型 134 個を探索の対象とした。初期状態はランダムに決定した。初期温度は `-O3` に初期状態の表すオプションを付加してプログラムを実行したときの実行時間、冷却率は 0.9、温度の冷却回数は 50 回、反復回数は 200 回とした。また、パラメータの値は最大 7 段階、実行時間の平均を求める際の実行回数は最大 5 とした。

実験結果を表 1 に示す。列 `-O0`、`-O3` が GCC の標準最適化オプションを付加したときの結果、列 `SA` がスイッチ型のみ探索結果で、列 `SA_P` がパラメータ型を含む探索結果である。パラメータ型を含む探索では、探索時間は増大しているが、`-O3` やスイッチ型のみ探索に比べ、対象のプログラムを平均約 3.9% 高速化できた。

表 1 実験結果

program	実行時間 [秒]				探索時間 [秒]	
	-O0	-O3	SA	SA_P	SA	SA_P
fasta.c	6.250	1.733	1.274	1.308	40385	45428
queen.c	2.064	0.561	0.444	0.430	14821	18022
bitcnts.c	6.935	1.828	1.267	1.270	43183	56384
nbody.c	14.007	2.265	1.976	1.615	62067	65053
fft.c	2.513	2.320	2.311	2.331	60154	62573

## 5 むすび

本研究では、パラメータ型オプションを含む最適化オプションセットの探索により、プログラムの実行を高速化する手法を提案した。今後の課題としては、メモリー使用量制限下での実行時間の削減等が考えられる。

### 参考文献

- [1] San-Chih Lin, Chi-Kuang Chang, and Nai-Wei Lin: "Automatic selection of GCC optimization options using a gene weighted genetic algorithm," in Proc. 13th ACSAC, pp. 1-8 (Aug. 2008).
- [2] 大城亮:関数毎指定及びパラメータ型オプションによるコンパイラの最適化オプションセット探索の拡張, 修士論文, 関西学院大学大学院理工学研究科 (Mar. 2013).